

*** NOTICES ***

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]An input memory measure which memorizes a data position and data size of input data in which a thing which was inputted from an input means, a printing means which prints print data, and an input means, and which print is possible, and its input data, A development control means which controls print data while developing memorized input data to print data based on a printing format, It has a print-data storing means which has three or more spread regions set up few rather than developing the 1-page contents, and stores print data in this spread region by turns one by one, Based on size of a spread region, a data position of input data, and data size, it is judged whether input data developed by print data is crossing a development control means to two or more spread regions, A memory controller for printing controlled repeatedly to transmit print data already developed from a spread region while gathering input data currently crossed to two or more spread regions in two or more spread regions and making it develop to a printing means.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention relates to the memory controller for printing in printing controlling processing of the printing deployment memory for developing a printing content in detail about the memory controller for printing.

[0002]

[Description of the Prior Art]Drawing 4 is a block diagram showing the composition of the conventional memory controller for printing. In the figure, 1 is CPU and performs printing controlling. 2A is a program / memory for data, and is because the program for carrying out printing controlling by CPU1 and input data, and control data are stored. 3A is a memory for printing deployment, and stores the data developed to compensate for the output format of a printer. 4 is a printer and prints the print data developed by the memory 3A for printing deployment. 5 is a keyboard and carries out the operational input for performing the input and printing of the data (alphabetic data/graphic data, etc.) for printing. 6 connects with each I/O Port of CPU1, a program / memory for data 2A, the memory 3A for printing deployment, the printer 4, and the keyboard 5, and is a bus line data / for addresses.

[0003]Drawing 5 is an explanatory view showing the print data for 1 page. As shown in the figure in a control device for printing like drawing 4, when printing the data for 1 page which consists of a portion of A-D, in order to reduce the memory space which a device needs, The memory 3A for printing deployment is constituted from capacity smaller than total unfolding capacity, and the way the division deployment divided into multiple times performs a printing job is known.

[0004]Drawing 6 is an explanatory view showing the example 1 of printing execution of drawing 5 by conventional technology. When there is the memory 3A for printing deployment which develops the data for one piece among A-D, processing is [in / the figure] possible as shown in drawing 5. First, the contents of A are developed in the memory 3A for printing deployment at the time t0-t1.

Then, the contents developed at the time t1-t2 are transmitted to the printer 4, and the contents of A are made to print with the printer 4.

Then, as soon as this transmission is completed, the contents which developed the contents of B which is the following contents in the memory 3A for printing deployment at the time t2-t3, and were developed at the time t3-t4 are transmitted to the printer 4. Henceforth, the contents of A-D are printed one by one with a printer about the contents of C, and the contents of D in a similar manner by performing the deployment to the memory 3A for printing deployment, and the output to the printer 4 one by one.

[0005]However, in the composition of performing deployment by the above memories 3A for printing deployment, and the output to the printer 4 by turns, during the deployment to a memory, it becomes the following print-data waiting time, and the problem which discontinuation of print operation produces in the middle of a page occurs in the printer side.

[0006]Therefore, in order to solve the above-mentioned problem, the memory 3A for printing deployment is divided into two pieces, and the parallel system which performs simultaneously output to the printer 4 of the data developed by one side and deployment to another side of the

following data is also held.

[0007]Namely, the transmission to the printer of the contents of A developed by one memory 3A for printing deployment, Although it is carried out one by one according to the capacity of the buffer memory actually built in the printer 4 to every [for one line] data (henceforth number-of-unit quantity data) and a printing job is carried out with the printer 4 to compensate for this transmission, It generally takes the time taken to print the above-mentioned number-of-unit quantity data in the printer 4 for a long time rather than the transfer time from the memory 3A for printing deployment memories to the printer 4.

[0008]For this reason, in the control action of CPU1, by the time it waits to complete the control for transmission of number-of-unit quantity data, and to complete the print operation corresponding to this with the printer 4 and starts transmission of the following number-of-unit quantity data, idle time will be produced. Then, the control action of CPU1 is switched to the time which makes the memory 3A for printing deployment develop the following data B in the meantime using this idle time.

[0009]Drawing 7 is an explanatory view showing the deployment/transfer cycle at the time of printing of a printer. As shown in the figure, by processing by the change of such control. By performing deployment to the memory 3A for printing deployment of the contents of B one by one at while transmitting the contents of A shown in drawing 5 to the printer 4 one by one and making them print for every number-of-unit quantity data, the idle time [transmission of number-of-unit quantity data is performed one by one] of a between, Transmission of the whole contents of A and deployment of the whole contents of B can carry out to parallel seemingly.

[0010]Drawing 8 is an explanatory view showing the example 2 of printing execution of drawing 5 by conventional technology. The above parallel processing explains the process in which the contents of said A-D are printed one by one, with reference to the figures. First, the contents of A are developed at the time t0-t1 by one memory 3A for printing deployment, and these developed contents are transmitted and printed by the printer 4 from the memory 3A for printing deployment at the time t1-t2. And processing which develops the contents of B in the memory 3A for printing deployment of another side in the meantime is performed simultaneously.

[0011]After this deployment is completed, the contents of B are transmitted and printed by the printer 4 at the time t2-t3, and the contents of C are newly developed by one memory (portion by which the contents of A were developed). Subsequently, transmission to the printer 4 of the contents of C and deployment to the memory 3A for printing deployment of the contents of D are performed at the time t3-t4. Finally, the contents of D are transmitted at the time t4-t5, and the contents of A-D are printed with the printer 4 one by one.

[0012]However, in the above-mentioned operation, transmission to the printer 4 of the contents of A started, for example at the time t1 and the deployment to the memory 3A for printing deployment of the contents of B are not necessarily completed simultaneously, and by which or the later one among these operations. The start time t2 of transmission of the contents of the following B and deployment of the contents of C is determined. If transmission of the contents of A is not completed, this is replaced with the contents of this A, cannot newly develop the contents of C, and it is because this cannot be transmitted to the printer 4, either, if deployment of the contents of B is not completed. Therefore, although expansion time increases, it can constitute from a small dosage and print operation can be made to perform rather than all the memory deployment dosages.

[0013]

[Problem(s) to be Solved by the Invention]However, in the conventional memory controller for printing, when developing data in the memory for printing deployment, it may happen that one collected data (for example, alphabetic data/graphic data) is developed in 2 steps.

[0014]Drawing 9 is an explanatory view showing an example of the print data over two memories for printing deployment. the character of A-D which quadrisected 1 page as shown in the figure — “ — *” — “ — the time of printing “ — a character “it is” — “ — about “, it did not go into one memory for deployment thoroughly, but two spread regions are straddled. However, let the size of alphabetic data be a thing smaller than the memory for printing deployment at this time.

[0015]Print operation of conventional technology is performed by the control method which was

mentioned above by drawing 4 – drawing 8. Here, a character “it is” will belong also to the portion of A, or the portion of B, and deployment will be performed in great numbers (however, only in case of portion which belongs to each field among characters that deployment is actually performed in the portions of A and B).

[0016]The following procedures are required as processing required in order to draw alphabetic data in the memory 3A for printing deployment. Methods of generally drawing a character include two, the method of having alphabetic data as a dot font, and the method of generating an outline font.

[0017](1): [In the case of a dot font]

(1-1) : it asks for the memory address in which the character is stored from a character code. (1-2) : a copy of data is performed to the memory which should draw from the memory in which alphabetic data is stored.

[0018](2): [In the case of an outline font]

(2-1) : it asks for the memory address in which the outline data of the character is stored from a character code.

(2-2) : the alphabetic data of the size needed based on outline data is created.

(2-3) : it transmits to the memory which should draw the created alphabetic data.

[0019]When not going into the memory which a part of character should draw in such a procedure, Since only the portion which actually draws is transmitted to a memory, in the case of a dot font, only processing of (1-2) and the part of the data volume in which processing of (2-3) should actually draw in the case of an outline font are transmitted to a memory.

[0020]However, as for processing of (1-1), and the case of an outline font, in the case of a dot font, processing of (1-1) and (1-2) will be performed twice. That is, in the example of drawing 4, when a character “it is” develops the contents of the A portion, the thing same twice when developing the contents of the portion of B will be performed. In particular, in the case of the outline font, the processing which creates alphabetic data from outline data started more mostly than time to actually transmit to a memory, and there was a problem used as very inefficient processing.

[0021]For example, as a method of reducing inefficiency, in the case of the outline font, the font data created from outline data is temporarily stored in a memory, and when needed, the so-called method of the font cache of obtaining font data from the portion is adopted.

[0022]However, since the capacity of cache memory was restricted also in such a method, when there were many alphabetic data/graphic data over the spread region whose number is two, there was a problem of being unable to respond.

[0023]This invention was made in consideration of the above situation, and abolishes the unnecessary processing time for printing deployment, and it is in providing the memory controller for printing which was rich in the cost performance which can reduce memory space as much as possible further.

[0024]

[Means for Solving the Problem]Drawing 1 is a block diagram showing basic constitution of this invention. In drawing 1, a memory controller for printing of this invention, The input memory measure 103 which memorizes a data position and data size of input data in which a thing which was inputted from the input means 101, the printing means 102 which prints print data, and the input means 101, and which print is possible, and its input data, The development control means 104 which controls print data while developing memorized input data to print data based on a printing format, It has the print-data storing means 105 which has three or more spread regions set up few rather than developing the 1-page contents, and stores print data in this spread region by turns one by one, Based on size of a spread region, a data position of input data, and data size, it is judged whether input data developed by print data is crossing the development control means 104 to two or more spread regions, Input data currently crossed to two or more spread regions is a memory controller for printing controlled repeatedly to transmit print data already developed from a spread region to the printing means 102 while collecting into two or more spread regions and making it develop.

[0025]In this invention, input devices, such as a keyboard, a tablet, or a pointing device, are used

as the input means 101. As the printing means 102, they are printers, such as a thermal transfer printer, an ink-jet printer, a laser beam printer, and a dot impact printer. As the input memory measure 103, the development control means 104, and the print-data storing means 105, CPU (central processing unit), ROM (read only memory), RAM (random access memory), It is convenient that a computer which consists of I/O Ports is used, and RAM in it and a floppy disk of an external storage, a hard disk, etc. are usually used as the input memory measure 103 and the print-data storing means 105.

[0026]

[Function]According to this invention, in drawing 1, if the input data which can be printed from the input means 101 is inputted, the inputted input data, its data position, and data size will be memorized by the input memory measure 103. Based on a printing format, by the development control means 104, print data develop and the memorized input data is stored in the spread region of the print-data storing means 105. The stored print data are printed by the printing means 102. Here, it has three or more spread regions set up few rather than the print-data storing means 105 develops the 1-page contents, and print data are stored in this spread region by turns one by one. Based on the size of a spread region, the data position of input data, and data size, it is judged whether when print data develop, the input data developed by print data is crossing the development control means 104 to two or more spread regions, The input data currently crossed to two or more spread regions is repeatedly controlled to transmit the print data already developed from the spread region to the printing means 102 while collecting into two or more spread regions and making it develop.

[0027]Therefore, print data are processed without making the dosage of development processing time or the memory for printing deployment increase, even if it is crossing three or more spread regions to two spread regions by using by turns.

[0028]

[Example]Hereafter, based on the example shown in a figure, this invention is explained in full detail. This invention is not limited by this.

[0029]Drawing 2 is a block diagram showing one example which applied this invention to the memory controller for printing. As for the memory for printing deployment, and 4, CPU and 2 are [a keyboard and 6] bus lines a printer and 5 a program / memory for data, and 3 1. About the composition of the memory controller for printing, although the contents differ among a program / memory 2 for data, the memory 3 for printing deployment, and the program / memory 2A for data of drawing 4, and the memory 3A for printing deployment, since they are fundamentally the same as the composition of a conventional example, explanation is omitted (refer to drawing 4).

[0030]Drawing 3 is a block diagram showing the internal configuration of the memory control part of the memory controller for printing in drawing 2. In the composition of drawing 3, CPU and 3 are memories for printing deployment which consist of a spread region of M1-M3, and 1 develops the data which should be outputted to a printer. The memory management unit with which 4 builds in a printer and 10 builds in a microprocessor, and 20 are memory control parts. In the case of a conventional example, in drawing 3, it becomes the composition which does not have a portion of M3, and a portion attached to it among the portions of M1-M3 of the three memories 3 for printing deployment, either.

[0031]In the memory control part 20 of the control device for printing of this invention, the next control is performed using three spread regions which consist of M1-M3 of the memory 3 for printing deployment, for example.

** When having transmitted the data of :M1 to the printer, develop the thing thoroughly contained in M2, and the thing over M2 and M3 to M2 and M3.

** When having transmitted the data of :M2 to the printer, develop the thing thoroughly contained in M3, and the thing over M3 and M1 to M3 and M1.

** When having transmitted the data of :M3 to the printer, develop the thing thoroughly contained in M1, and the thing over M1 and M2 to M1 and M2.

In order to repeat these three unfolding operation by turns, between CPU1 which carries out printing controlling, the deployment memory 3 (M1, M2M3) for printing, and the printer 4, the

address and the buffer which controls the connected state of data are formed.

[0032]The address signal and M1 D-M3D which M1-M3 give M1 A-M 3A, respectively show the data of M1-M3, respectively. CA expresses the address control signals (and read/write etc.) from CPU1, and CD expresses the data of CPU1. PD shows the data outputted to the printer 4. BF1-BF3, BF1 C-BF3C, and BF1 P-BF3P are buffers which control the connected state of a signal wire. When ON1-ON3 become active, BF1-BF3 will be in switch-on, respectively. When BF1 is in switch-on, CA and M1A are connected, when BF2 is in switch-on, CA and M2A are connected, and when BF3 is in switch-on, CA and M3A are connected.

[0033]If ON1 C-ON3C becomes active, BF1 C-BF3C will be in switch-on, and connection of the signal wire which each has managed is controlled. If ON1 P-ON3P becomes still more active, BF1 P-BF3P will be in switch-on, respectively, and same control will be performed.

[0034]It is determined whether the memory management unit 10 connects M1 to M3 with CPU1 or the printer 4 by creating the control signal which controls opening and closing of the buffer explained above. M1 to M3 shall be seen from CPU1, respectively, and shall be assigned to the separate address.

[0035]Drawing 10 is an explanatory view showing the address quota method of the trichotomized memory for printing deployment. For example, as shown in drawing 10, assigning M2 to the address 2000H - 2FFFH, and assigning M3 to the address 1000H - 1FFFH for M1 at 3000H - 3FFFH is shown. Here, the hexadecimal notation shall show the 1000th street in 1000H.

[0036]Thus, if an address is assigned to the memory 3 for printing deployment, the memory management unit 10 can judge which memory is accessed under supervising the address signal from CPU1. For example, M2 will be accessed, if 1500H street is accessed and 2400H street is accessed to M1.

[0037]It can be judged by whether the control signal showing the operating state of CPU1 is a read state, or it is in a light state whether the memory 3 for printing deployment is connected with CPU1, or the memory 3 is connected with the printer 4. That is, since it expresses with the printer 4 from the memory that it is in the state which outputs data if it is connection and a read state about CPU1 and the memory 3, since it expresses that it is a writing state from CPU1 to a memory if it is in a light state, the memory 3 is connected with the printer 4.

[0038]For example, if it turns out that it is in the light state of the data to 1200H street, CPU1 and M1 will be connected, and M3 will be connected with the printer 4 if it turns out that it is in the light state of the data to 3400H street.

[0039]Drawing 11 is an explanatory view showing the state of the control signal of the address sent to the memory for printing deployment, and CPU. Here, when connecting CPU1 and M1, only two buffers of BF1 and BF1C are made into a connected state (two of ON1 and ON1C are made into an active state), and CA, M1A, CD, and M1D are connected. Thus, it can judge which buffer should be made into a connected state under supervising the state of CPU1.

[0040]Control at the time of printing is performed as follows using the memory control part 20 of such composition. That is, the thing to the conventional printing controlling method of drawing 8 shows drawing 12.

[0041]At this time, the contents to print shall consist of a portion of A-D, as shown in drawing 5. When there are some in which data exists over two, A and B, the thing belonging to the B side of that is defined as AB. BC and the thing which exists over B and C also about CD shall be defined similarly.

[0042]Drawing 12 is an explanatory view showing the example of printing execution of drawing 9 of this invention. As shown in drawing 12, first, in the time t0-t1, the contents of A are developed in the memory M1 for printing deployment, and printing controlling develops the contents of AB to M2 further. In the time t1-t2, the contents of (B-AB) are developed in the memory M2 for printing deployment, and the contents of BC are developed to M3. Here, (B-AB) expresses the thing excluding the contents of AB from the contents of B. In parallel with this, the contents of A are transmitted to a printer from M1.

[0043]The time t2 is determined by the one where ending among one of operations is later at this time. Although the contents of C are developed to M3 and it develops the contents of CD to M1 in the time t2-t3, if M1 is not vacant as for this at this time, it is because deployment of the

[0044]here — the contents of CD — for example, some alphabetic data — the character in drawing 9 — " — it expresses that it is what is contained in the portion of D among ". That is, since the remaining portion is contained in the contents of C, it is necessary to wait for it to be inefficient-like to completely develop the contents of C and CD independently, therefore for M1 to be vacant as for it. At this time, it is also possible to make it develop previously without waiting for M1 to be vacant about what is contained only in the portion of C like the character "obtain" in drawing 9. However, this example shows what is not being performed.

[0045] Furthermore, in the time t2-t3, the contents of (C-BC) are developed to M3, and the contents of CD are developed to M1. It is parallel to it and the contents of B are transmitted to a printer from M2. In the time t3-t4, the contents of (D-CD) are developed to M1. In parallel with it, the contents of C are transmitted to a printer from M3. Finally, in the time t4-t5, the contents of D will be transmitted to a printer from M1, and all the printing jobs will be completed.

[0046] Next, how to judge whether alphabetic data belongs to two spread regions is explained.

[0047]Drawing 13 is an explanatory view showing the controlling method of the print point of print data. As shown in the figure, coordinates management of the two dimensions of x and y shall be beforehand carried out to print data. Here, an x-coordinate is taken as the direction which increases rightward, and the direction which a y-coordinate increases downward. The number of the things equivalent to one spread region shall be 300, and there shall be 800 dots in a lengthwise direction as a whole in a lengthwise direction at 1200 dots and a transverse direction. In such a coordinate system, it manages on the position which each alphabetic data should draw, and a table as show drawing 14 character size.

[0048] Drawing 14 is an explanatory view showing the management table of drawing data (print data). In the figure, as it is in drawing 9, as for the position which should draw, upper left coordinates and character size express the dot number of length and each width. For example, a character "**" is = (100, 50) as a drawing position (x, y), and it is shown that character sizes are vertical dot number =150 and horizontal dot number =150 dot. other characters — " — it is — " — " — ***** — it is shown similarly. It can such be searched whether alphabetic data is contained in which spread region using the management table, respectively.

[0049]Drawing 15 is a flow chart which shows the development processing of the print data of the memory controller for printing in this invention. In the flow chart of the figure, the drawing data over the two memories 3 for printing deployment is searched, and the method to a memory to develop is explained. Step S101, S102: Compare the size relation of the Y coordinate of the starting position of a spread region, and a drawing Y coordinate, and judge whether the drawing (character) data is contained in the present spread region. For example, in drawing 13 and the example of drawing 14, when the present spread region should develop the contents of A, since it is start Y coordinate=0 of the field of B, and end Y coordinate=299, about a character "**", The drawing coordinate of a character "*" = since 50 are contained in the meantime, it turns out that it is data which should draw now, and shifts from them to the processing after Step S103. Here, when it is judged that it does not go into the field which should be developed, it shifts to processing of S106.

[0050]Step S103: Judge whether drawing data is being brought into the next spread region. That is, the size relation of the vertical dot number of the Y coordinate + character size of drawing data is compared with the end Y coordinate of the present spread region. Drawing data will be brought into the next spread region if the value of the end Y coordinate of a spread region is smaller. For example, when the present spread region is in the position which should develop the contents of A, about a character "it is." The vertical dot number of the Y coordinate + alphabetic data of drawing end position = since it is set to $200+150=350$ and this becomes larger than 300 which is an end Y coordinate of the field of A, it turns out that a character "it is" is contained also in B which is the next field. It progresses to Step S105 at this time. If it turns out that drawing data is not including the following memory for deployment at Step S103, it will progress to Step S104.

[0051]Step S104: If drawing data is not including the following memory for deployment, develop drawing data in the present memory for deployment.

Step S105: Develop drawing data using the present spread region and the spread region which should be used for the next.

Step S106: Judge whether processing was completed about all drawing data. Step S107: If there is the remaining data, in order to repeat the processing from S101 and to perform it, move a processing object to the following drawing data. Such a drawing management table is storable in the program / memory 2 for data in drawing 2. Thus, deployment of the data crossed to two spread regions is performed.

[0052]It is possible to control the portion of a printer similarly not only as memory controllers for printing, such as a printer, as an application in the file control system replaced with a disk, the network control system replaced with an interface with the exterior, etc., for example.

[0053]

[Effect of the Invention]According to this invention, since it is not large about the further needed increase in memory space, either, without making the expansion time to the memory of print data increase when printing, control of the memory for printing excellent in cost performance can be performed.

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-19652

(43)公開日 平成 6 年(1994) 1 月28日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 3/12	N			
B 4 1 J 2/505				
2/485				
		9211-2C	B 4 1 J 3/ 10	1 0 1 Z
		8804-2C	3/ 12	C
審査請求 未請求 請求項の数 1 (全 14 頁) 最終頁に続く				

(21)出願番号 特願平4-173496

(22)出願日 平成 4 年(1992) 6 月30日

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72)発明者 坂本 辰彦

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72)発明者 向殿 充浩

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(72)発明者 竹田 均

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(74)代理人 弁理士 野河 信太郎

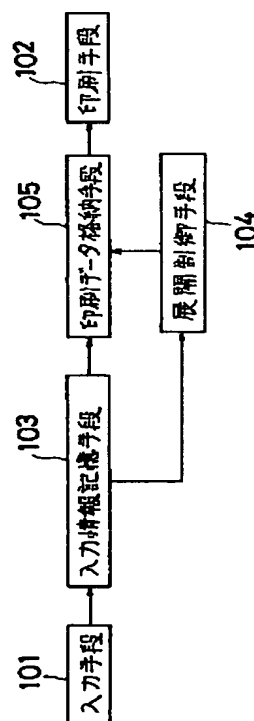
最終頁に続く

(54)【発明の名称】 印刷用メモリ制御装置

(57)【要約】

【構成】 入力手段と、印刷データを印刷する印刷手段と、入力手段から入力された印刷することが可能な入力データとその入力データのデータ位置およびデータサイズを記憶する入力情報記憶手段と、入力データを印刷フォーマットに基づいて印刷データに展開するとともに印刷データを制御する展開制御手段と、三つ以上の展開領域を有しこの展開領域に順次交互に印刷データを格納する印刷データ格納手段を備え、展開制御手段は入力データが二つ以上の展開領域に渡っているかどうかを展開領域のサイズと入力データのデータ位置およびデータサイズに基づいて判断し、二つ以上の展開領域に渡っている入力データは二つ以上の展開領域にまとめて展開するように構成される。

【効果】 印刷を行う際、メモリ容量の増加を押さえ、さらにメモリへの印刷データの展開時間を増加させないため、コストパフォーマンスに優れた印刷制御ができる。



【特許請求の範囲】

【請求項 1】 入力手段と、印刷データを印刷する印刷手段と、入力手段から入力された印刷することが可能な入力データとその入力データのデータ位置およびデータサイズを記憶する入力情報記憶手段と、記憶された入力データを印刷フォーマットに基づいて印刷データに展開するとともに印刷データを制御する展開制御手段と、1ページのコンテンツを展開するより少なく設定されている三つ以上の展開領域を有しこの展開領域に順次交互に印刷データを格納する印刷データ格納手段とを備え、展開制御手段は印刷データに展開される入力データが二つ以上の展開領域に渡っているかどうかを展開領域のサイズと入力データのデータ位置およびデータサイズに基づいて判断し、二つ以上の展開領域に渡っている入力データは二つ以上の展開領域にまとめて展開するようにするとともに展開領域からすでに展開されている印刷データを印刷手段に転送するよう繰り返し制御する印刷用メモリ制御装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は印刷用メモリ制御装置に関し、詳しくは、印刷内容の展開を行うための印刷展開メモリの印刷制御処理における印刷用メモリ制御装置に関する。

【0002】

【従来の技術】 図 4 は従来の印刷用メモリ制御装置の構成を示すブロック図である。同図において、1 は CPU であり、印刷制御を行うものである。2 A はプログラム／データ用メモリであり、CPU 1 により印刷制御するためのプログラム及び入力データ、制御データを格納するためのものである。3 A は印刷展開用メモリであり、プリンタの出力フォーマットに合わせて展開されたデータの格納を行うものである。4 はプリンタであり、印刷展開用メモリ 3 A に展開された印刷データを印刷するものである。5 はキーボードであり、印刷するためのデータ（文字データ／図形データ等）の入力と印刷を実行するための操作入力をするものである。6 は CPU 1、プログラム／データ用メモリ 2 A、印刷展開用メモリ 3 A、プリンタ 4、キーボード 5 の各 I/O ポートと接続し、データ／アドレス用のバスラインである。

【0003】 図 5 は 1 ページ分の印刷データを示す説明図である。図 4 のような印刷用制御装置において、同図に示すように、A～D の部分からなる 1 ページ分のデータの印刷を行うとき、装置が必要とするメモリ容量を削減するため、全展開容量よりも少ない容量で印刷展開用メモリ 3 A を構成し、複数回に分けた分割展開により印刷処理を行う方法が知られている。

【0004】 図 6 は従来技術による図 5 の印刷実行例 1 を示す説明図である。同図において、例えば、A～D の内、1 個分のデータを展開するだけの印刷展開用メモリ

3 A があつた場合には、図 5 に示すように、処理を行うことが可能である。まず、時刻 $t_0 \sim t_1$ で A の内容を印刷展開用メモリ 3 A に展開し、その後、時刻 $t_1 \sim t_2$ で展開した内容をプリンタ 4 に転送して、プリンタ 4 で A の内容を印刷させる。続いて、この転送が終了しだい、時刻 $t_2 \sim t_3$ で次の内容である B の内容を印刷展開用メモリ 3 A に展開し、時刻 $t_3 \sim t_4$ で展開した内容をプリンタ 4 に転送する。以降、同様に C の内容、D の内容について、印刷展開用メモリ 3 A への展開、プリンタ 4 への出力を順次行うことによって、プリンタで A～D の内容が順次印刷される。

【0005】 しかし、上記のような印刷展開用メモリ 3 A での展開と、プリンタ 4 への出力を交互に行う構成では、メモリへの展開中は、プリンタ側では次の印刷データ待ち時間になって、ページの途中で印刷動作の中断が生じる問題が発生する。

【0006】 従って、上記の問題を解決するために、印刷展開用メモリ 3 A を 2 個に分け、一方で展開されたデータのプリンタ 4 への出力と、次のデータの他方への展開を同時に行う並列方式も行われている。

【0007】 すなわち、一方の印刷展開用メモリ 3 A に展開された A の内容のプリンタへの転送は、実際にはプリンタ 4 に内蔵されているバッファメモリの容量に合わせて、例えば、1 行分のデータ（以下、単位数量データという）毎に順次行われ、この転送に合わせてプリンタ 4 で印刷処理されるが、プリンタ 4 での上記単位数量データを印刷するのに要する時間は、印刷展開メモリ用メモリ 3 A からプリンタ 4 への転送時間よりも一般的に長くかかる。

【0008】 このため、CPU 1 での制御動作には、単位数量データの転送のための制御を完了し、これに対応する印刷動作がプリンタ 4 で完了するのを待って次の単位数量データの転送を開始するまでに空き時間を生じる。そこで、この空き時間を利用して、この間、CPU 1 での制御動作を、次のデータ B を印刷展開用メモリ 3 A に展開させる時間に切り換えるものである。

【0009】 図 7 はプリンタの印刷時における展開／転送サイクルを示す説明図である。同図に示すように、このような制御の切り換えによる処理によって、図 5 に示す A の内容を単位数量データ毎にプリンタ 4 に順次転送して印刷させるとともに単位数量データの転送が順次行われる間の空き時間に、B の内容の印刷展開用メモリ 3 A への展開も順次行うことによって、A の内容全体の転送と、B の内容全体の展開とが見かけ上並列に行うことができる。

【0010】 図 8 は従来技術による図 5 の印刷実行例 2 を示す説明図である。上記のような並列処理によって、前記 A～D の内容を順次印刷する過程について、同図を参照して説明する。まず、A の内容が時刻 $t_0 \sim t_1$ で一方の印刷展開用メモリ 3 A に展開され、この展開され

10

20

30

40

50

3

た内容は、時刻 $t_1 \sim t_2$ で印刷展開用メモリ 3 A からプリンタ 4 に転送されて印刷される。そして、この間に B の内容を他方の印刷展開用メモリ 3 A に展開する処理が同時に行われる。

【0011】この展開が終了すると、時刻 $t_2 \sim t_3$ で B の内容がプリンタ 4 に転送されて印刷されると共に、一方のメモリ (A の内容が展開されていた部分) に C の内容が新たに展開される。次いで、時刻 $t_3 \sim t_4$ で、C の内容のプリンタ 4 への転送と、D の内容の印刷展開用メモリ 3 A への展開が行われる。最後に時刻 $t_4 \sim t_5$ で、D の内容が転送されて、A ~ D の内容が順次プリンタ 4 で印刷される。

【0012】しかし、上記動作においても、例えば時刻 t_1 で開始された A の内容のプリンタ 4 への転送と、B の内容の印刷展開用メモリ 3 A への展開が同時に終了するわけではなく、これらの動作の内、どちらか遅い方によって、次の B の内容の転送と C の内容の展開との開始時刻 t_2 が決定される。これは、A の内容の転送が完了していなければ、この A の内容に代えて、新たに C の内容の展開を行えないと共に、B の内容の展開が終了していなければこれをプリンタ 4 へ転送することもできないからである。従って、展開時間が増加するがメモリ全展開用量よりも少ない用量で構成して印刷動作を行わせることができる。

【0013】

【発明が解決しようとする課題】しかしながら、従来の印刷用メモリ制御装置において、印刷展開用メモリにデータを展開するとき、1 つのまとまったデータ (例えば、文字データ/図形データ) が 2 回に分けて展開されることが起こり得る。

【0014】図 9 は二つの印刷展開用メモリに渡る印刷データの一例を示す説明図である。同図に示すように、1 ページ分を 4 分割した A ~ D の文字 “あ” ~ “か” を印刷するとき、文字 “い”、“お” については 1 個の展開用メモリには完全に入らず、2 つの展開領域にまたがっている。ただし、このとき、文字データの大きさは印刷展開用メモリより小さいものとする。

【0015】従来技術の印刷動作は、図 4 ~ 図 8 で上述したような制御方法で行われる。ここで、文字 “い” は A の部分にも B の部分にも属し、両方で展開が行われることになる (ただし、実際に A、B の部分で展開が行われるのは、文字の内それぞれの領域に属している部分のみ)。

【0016】文字データを印刷展開用メモリ 3 A に描画するために必要な処理として以下のような手順が必要である。一般的に文字を描画する方法として、文字データをドットフォントとしてもつ方法、及びアウトラインフォントを発生させる方法の 2 つがある。

【0017】(1) : [ドットフォントの場合]

(1-1) : 文字コードから、その文字が格納されている

4

メモリアドレスを求める。(1-2) : 文字データが格納されているメモリから、描画すべきメモリに対してデータのコピーを行う。

【0018】(2) : [アウトラインフォントの場合]
(2-1) : 文字コードから、その文字のアウトラインデータが格納されているメモリアドレスを求める。

(2-2) : アウトラインデータを基に、必要とする大きさの文字データを作成する。

(2-3) : 作成された文字データを描画すべきメモリに転送する。

【0019】このような手順において、文字の一部しか描画すべきメモリに入っていない場合には、実際に描画する部分しかメモリに転送しないので、ドットフォントの場合は (1-2) の処理、アウトラインフォントの場合は (2-3) の処理が実際に描画すべきデータ量の分だけメモリに転送される。

【0020】しかし、ドットフォントの場合は (1-1) の処理、アウトラインフォントの場合は (1-1) 及び (1-2) の処理が 2 回行われることになる。つまり、図 4 の例では文字 “い” は A 部分の内容を展開するとき、及び B の部分の内容を展開する時の 2 回同じことが行われてしまう。特に、アウトラインフォントの場合にはアウトラインデータから文字データを作成する処理が実際にメモリに転送する時間より多くかかり、非常に非効率的な処理となっている問題があった。

【0021】例えば、非効率性を低減させる方法として、アウトラインフォントの場合、アウトラインデータから作成されたフォントデータを一時的にメモリに格納し、必要となったときにその部分からフォントデータを得るといいうゆるフォントキャッシュの方法が採用されている。

【0022】しかしながら、このような方法においてもキャッシュメモリの容量が制限されるため、二つの展開領域にまたがる文字データ/図形データが多い場合には対応できないなどの問題があった。

【0023】この発明は以上の事情を考慮してなされたもので、印刷展開のための不要な処理時間をなくし、さらにメモリ容量をできるだけ低減させることが可能なコストパフォーマンスに富んだ印刷用メモリ制御装置を提供することにある。

【0024】

【課題を解決するための手段】図 1 はこの発明の基本構成を示すブロック図である。図 1 において、この発明の印刷用メモリ制御装置は、入力手段 101 と、印刷データを印刷する印刷手段 102 と、入力手段 101 から入力された印刷することが可能な入力データとその入力データのデータ位置およびデータサイズを記憶する入力情報記憶手段 103 と、記憶された入力データを印刷フォーマットに基づいて印刷データに展開するとともに印刷データを制御する展開制御手段 104 と、1 ページの内

容を展開するより少なく設定されている三つ以上の展開領域を有しこの展開領域に順次交互に印刷データを格納する印刷データ格納手段 105 とを備え、展開制御手段 104 は印刷データに展開される入力データが二つ以上の展開領域に渡っているかどうかを展開領域のサイズと入力データのデータ位置およびデータサイズに基づいて判断し、二つ以上の展開領域に渡っている入力データは二つ以上の展開領域にまとめて展開するようにするとともに展開領域からすでに展開されている印刷データを印刷手段 102 に転送するよう繰り返し制御する印刷用メモリ制御装置である。

【0025】なお、この発明において、入力手段 101 としては、キーボード、タブレット、あるいはポインティングデバイス等の入力装置が用いられる。印刷手段 102 としては、熱転写プリンタ、インクジェットプリンタ、レーザプリンタ、ドットインパクトプリンタ等の印刷装置である。入力情報記憶手段 103、展開制御手段 104、印刷データ格納手段 105 としては、CPU (中央処理装置)、ROM (リード・オンリ・メモリ)、RAM (ランダム・アクセス・メモリ)、I/O

【0026】

【作用】この発明によれば、図 1 において、入力手段 101 から印刷することが可能な入力データが入力されると、入力された入力データとそのデータ位置およびデータサイズが入力情報記憶手段 103 に記憶される。記憶された入力データは印刷フォーマットに基づいて展開制御手段 104 によって印刷データに展開され、印刷データ格納手段 105 の展開領域に格納される。格納された印刷データは印刷手段 102 により印刷される。ここで、印刷データ格納手段 105 は 1 ページの内容を展開するより少なく設定されている三つ以上の展開領域を有し、この展開領域に順次交互に印刷データが格納される。印刷データに展開される際に、展開制御手段 104 は印刷データに展開される入力データが二つ以上の展開領域に渡っているかどうかを展開領域のサイズと入力データのデータ位置およびデータサイズに基づいて判断し、二つ以上の展開領域に渡っている入力データは二つ以上の展開領域にまとめて展開するようにするとともに展開領域からすでに展開されている印刷データを印刷手段 102 に転送するよう繰り返し制御する。

【0027】従って、三つ以上の展開領域を交互に利用することで、二つの展開領域に渡っていても展開処理時間や印刷展開用メモリの用量を増加させないで印刷データが処理される。

【0028】

【実施例】以下、図に示す実施例に基づいてこの発明を詳述する。なお、これによってこの発明は限定されるものでない。

【0029】図 2 はこの発明を印刷用メモリ制御装置に適用した一実施例を示すブロック図である。1 は CPU、2 はプログラム/データ用メモリ、3 は印刷展開用メモリ、4 はプリンタ、5 はキーボード、6 はバスラインである。印刷用メモリ制御装置の構成については、プログラム/データ用メモリ 2、印刷展開用メモリ 3 と図 4 のプログラム/データ用メモリ 2A、印刷展開用メモリ 3A とは内容が異なるが基本的に従来例の構成と同じであるので説明を省略する (図 4 参照)。

【0030】図 3 は図 2 における印刷用メモリ制御装置のメモリ制御部の内部構成を示すブロック図である。図 3 の構成において、1 は CPU、3 は M1~M3 の展開領域からなる印刷展開用メモリであり、プリンタに出力すべきデータを展開する。4 はプリンタ、10 はマイクロプロセッサを内蔵するメモリ管理装置、20 はメモリ制御部である。図 3 において、従来例の場合は、三つの印刷展開用メモリ 3 の M1~M3 の部分の内、M3 の部分とそれに付随する部分もない構成となる。

【0031】この発明の印刷用メモリ制御装置のメモリ制御部 20 では、例えば、印刷展開用メモリ 3 の M1~M3 からなる 3 個の展開領域を用いて次の制御を行う。

①: M1 のデータをプリンタに転送しているときには、M2 に完全に含まれるもの、及び M2 と M3 にまたがるものを M2、及び M3 に展開する。

②: M2 のデータをプリンタに転送しているときには、M3 に完全に含まれるもの、及び M3 と M1 にまたがるものを M3、及び M1 に展開する。

③: M3 のデータをプリンタに転送しているときには、M1 に完全に含まれるもの、及び M1 と M2 にまたがるものを M1、及び M2 に展開する。

これら、三つの展開動作を交互に繰り返すため、印刷制御する CPU 1 と、印刷用展開メモリ 3 (M1、M2、M3) と、プリンタ 4 との間にはアドレス、およびデータの接続状態を制御するバッファが設けられている。

【0032】M1A~M3A はそれぞれ M1~M3 の与えるアドレス信号、M1D~M3D はそれぞれ M1~M3 のデータを示している。さらに、CA は CPU 1 からアドレス (及びリード・ライト等の) 制御信号、CD は CPU 1 のデータを表している。PD はプリンタ 4 に出力するデータを示す。BF1~BF3、BF1C~BF3C、BF1P~BF3P は信号線の接続状態を制御するバッファである。ON1~ON3 がアクティブになったとき、それぞれ BF1~BF3 が導通状態となる。BF1 が導通状態となったときには CA と M1A が接続され、BF2 が導通状態となったときには CA と M2A が接続され、BF3 が導通状態となったときには CA と M3A が接続される。

【0033】また、ON1C～ON3CがアクティブとなるとBF1C～BF3Cが導通状態となり、それぞれが管理している信号線の接続を制御する。さらにON1P～ON3Pがアクティブとなると、BF1P～BF3Pがそれぞれ導通状態となり同様の制御が行われる。

【0034】メモリ管理装置10は以上で説明したバッファの開閉を制御する制御信号を作成するものであり、M1からM3をCPU1またはプリンタ4と接続するかを決定する。M1からM3はそれぞれCPU1から見て

10 別々のアドレスに割り付けられているものとする。
【0035】図10は3分割された印刷展開用メモリのアドレス割り当て方法を示す説明図である。例えば、図10に示すように、M1をアドレス1000H～1FFFHに、M2をアドレス2000H～2FFFFHに、M3を3000H～3FFFFHに割り当てていることを示している。ここで、1000Hとは16進で1000番地を示すものとする。

【0036】このように印刷展開用メモリ3にアドレスを割り当てると、メモリ管理装置10はCPU1からのアドレス信号を監視することでどのメモリがアクセスされているかを判断することができる。例えば、1500H番地がアクセスされていればM1が、2400H番地がアクセスされていればM2がアクセスされていることになる。

【0037】さらに、CPU1と印刷展開用メモリ3を接続するか、またはプリンタ4とメモリ3を接続するかはCPU1の動作状態を表す制御信号がリード状態であるか、ライト状態であるかによって判断することができる。即ち、ライト状態であれば、CPU1からメモリへの書き込み状態であることを表しているため、CPU1とメモリ3を接続、リード状態であれば、メモリからプリンタ4にデータを出力する状態であることを表しているため、プリンタ4とメモリ3を接続する。

【0038】例えば、1200H番地へのデータのライト状態であることが分かれば、CPU1とM1を接続し、3400H番地へのデータのライト状態であることが分かれば、プリンタ4とM3を接続する。

【0039】図11は印刷展開用メモリに送るアドレスとCPUの制御信号の状態を示す説明図である。ここで、CPU1とM1を接続するときには、BF1、BF1Cの二つのバッファのみを接続状態とし（ON1、ON1Cの二つをアクティブ状態とする）、CAとM1A、CDとM1Dを接続する。このように、CPU1の状態を監視することで、どのバッファを接続状態とすればよいかを判断することができる。

【0040】このような構成のメモリ制御部20を用いて、印刷時の制御を以下のように行う。つまり、図8の従来の印刷制御方法に対するものが、図12に示すものである。

【0041】このとき、印刷する内容は図5に示したよ

うにA～Dの部分からなるものとする。さらに、AとBの2つに渡ってデータが存在するものがある場合、そのB側に属するものをABとして定義する。同様にBC、CDについてもBとCに渡って存在するものに関して定義するものとする。

【0042】図12はこの発明の図9の印刷実行例を示す説明図である。図12に示すように、印刷制御は、まず、時刻t0～t1において、印刷展開用メモリM1にAの内容を、さらにM2にABの内容を展開する。時刻t1～t2においては、(B-AB)の内容を印刷展開用メモリM2に、BCの内容をM3に展開する。ここで、(B-AB)とは、Bの内容からABの内容を除いたものを表している。これと平行してM1からプリンタにAの内容を転送する。

【0043】このとき、いずれかの動作の内、終了するのが遅い方によって、時刻t2が決定される。これは、時刻t2～t3においてCの内容をM3に、CDの内容をM1に展開するが、このときM1が空いていなければ、内容の展開が不可能なためである。

20 【0044】ここで、CDの内容は例えば、文字データの一部図9における文字“お”の内でDの部分に含まれるものであることを表している。つまり、残りの部分はCの内容に含まれているので、CとCDの内容を全く独立に展開することは非効率的で、そのためM1の空くのを待つ必要がある。このとき、図9における文字“え”のようにCの部分にしか含まれないものについては、M1の空くのを待たないで先に展開するようにすることも可能である。しかし、この例では、行っていないものを示している。

30 【0045】さらに時刻t2～t3においては、(C-BC)の内容をM3に、CDの内容をM1に展開する。それと平行して、Bの内容をM2からプリンタに転送する。時刻t3～t4においては、(D-CD)の内容をM1に展開する。それと平行してCの内容をM3からプリンタに転送する。最後に時刻t4～t5において、Dの内容をM1からプリンタに転送し、全ての印刷処理が終了することになる。

【0046】次に、文字データが2つの展開領域に属しているか否かを判定する方法について説明する。

40 【0047】図13は印刷データの印刷位置の管理方法を示す説明図である。同図に示すように、印刷データに対して、あらかじめx、yの2次元の座標管理がされているものとする。ここで、x座標は右向きに増加する方向、y座標が下向きに増加する方向とする。さらに、1個の展開領域に相当するものが縦方向に300ドットであり、全体として縦方向に1200ドット、横方向に800ドットあるものとする。このような座標系において、各文字データの描画すべき位置、および文字サイズを図14に示すようなテーブルで管理する。

50 【0048】図14は描画データ（印刷データ）の管理

テーブルを示す説明図である。同図において、図 9 中にあるように、描画すべき位置とは左上の座標、文字サイズは縦、横それぞれのドット数を表している。例えば、文字“あ”は描画位置として $(x, y) = (100, 50)$ であり、文字サイズが縦ドット数 = 150、横ドット数 = 150 ドットであることを示している。他の文字“い”～“か”についても同様に示されている。このような、管理テーブルを用いて文字データがそれぞれの展開領域に含まれているかを検索することができる。

【0049】図 15 はこの発明における印刷用メモリ制御装置の印刷データの展開処理を示すフローチャートである。同図のフローチャートにおいて、二つの印刷展開用メモリ 3 に渡る描画データを検索しメモリへの展開する方法を説明する。ステップ S101、S102：展開領域の開始位置の Y 座標と描画 Y 座標との大小関係を比較し、その描画（文字）データが現在の展開領域に入っているかどうかを判断する。例えば、図 13、及び図 14 の例において、現在の展開領域が A の内容を展開すべきであるとき、文字“あ”については B の領域の開始 Y 座標 = 0、終了 Y 座標 = 299 であるので、文字“あ”の描画座標 = 50 はこの間に含まれるため、現在描画すべきデータであることが分かり、ステップ S103 以降の処理に移る。ここで、もし、展開すべき領域に入っていないと判断された場合には S106 の処理に移る。

【0050】ステップ S103：描画データが次の展開領域に渡っているかどうかを判断する。即ち、現在の展開領域の終了 Y 座標と、描画データの Y 座標 + 文字サイズの縦ドット数の大小関係を比較する。もし、展開領域の終了 Y 座標の値の方が小さければ、描画データが次の展開領域に渡っていることになる。例えば、現在の展開領域が A の内容を展開すべき位置にあるときに、文字“い”については、描画終了位置の Y 座標 + 文字データの縦ドット数 = $200 + 150 = 350$ となり、これが A の領域の終了 Y 座標である 300 より大きくなるため、文字“い”は次の領域である B の中にも含まれることが分かる。このときはステップ S105 に進む。もし、ステップ S103 で描画データが次の展開用メモリに渡っていないことがわかれば、ステップ S104 に進む。

【0051】ステップ S104：描画データが次の展開用メモリに渡っていなければ、現在の展開用メモリに描画データを展開する。

ステップ S105：現在の展開領域と次に使用すべき展開領域を利用して、描画データの展開を行う。

ステップ S106：全ての描画データについて処理が終了したかどうかを判断している。ステップ S107：もし、残っているデータがあれば、S101 からの処理を繰り返し行うために、次の描画データに処理対象を移す。なお、このような描画管理テーブルは、図 2 におけるプログラム／データ用メモリ 2 内に格納しておくことができる。このようにして、二つの展開領域へ渡るデータの

展開が行われる。

【0052】また、さらに応用例として、プリンタなどの印刷用メモリ制御装置としてだけでなく、例えば、プリンタの部分をディスクに置き換えたファイル制御システム、外部とのインターフェースに置き換えたネットワーク制御システム等においても同様に制御することが可能である。

【0053】

【発明の効果】この発明によれば、印刷を行う際に、印刷データのメモリへの展開時間を増加させることなく、さらに必要となるメモリ容量の増加についても多くないため、コストパフォーマンスに優れた印刷用メモリの制御ができる。

【図面の簡単な説明】

【図 1】この発明の基本構成を示すブロック図。

【図 2】この発明を印刷用メモリ制御装置に適用した一実施例の構成を示すブロック図。

【図 3】図 2 における印刷用メモリ制御装置のメモリ制御部の内部構成を示すブロック図。

【図 4】従来の印刷用メモリ制御装置の構成を示すブロック図。

【図 5】1 ページ分の印刷データを示す説明図。

【図 6】従来技術による図 5 の印刷実行例 1 を示す説明図。

【図 7】プリンタの印刷時における展開／転送サイクルを示す説明図。

【図 8】従来技術による図 5 の印刷実行例 2 を示す説明図。

【図 9】二つの印刷展開用メモリに渡る印刷データの一例を示す説明図。

【図 10】3 分割された印刷展開用メモリのアドレスの割り当て方法を示す説明図。

【図 11】印刷展開用メモリに送るアドレスと CPU の制御信号の状態を示す説明図。

【図 12】この発明における図 8 の印刷実行例を示す説明図。

【図 13】印刷データの印刷位置の管理方法を示す説明図。

【図 14】描画データ（印刷データ）の管理テーブルを示す説明図。

【図 15】この発明における印刷用メモリ制御装置の印刷データの展開処理を示すフローチャート。

【符号の説明】

1 CPU（中央処理装置）

2 プログラム／データ用メモリ

3 印刷展開用メモリ

4 プリンタ

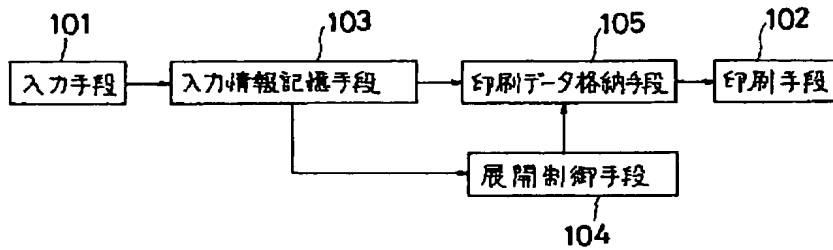
5 キーボード

6 バスライン

10 メモリ管理装置

20 メモリ制御部

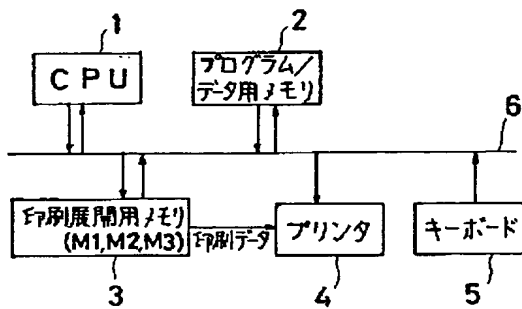
【図 1】



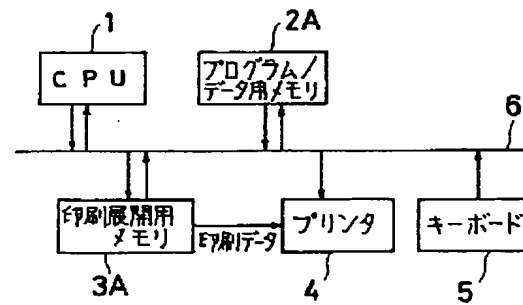
【図 5】

A
B
C
D

【図 2】



【図 4】



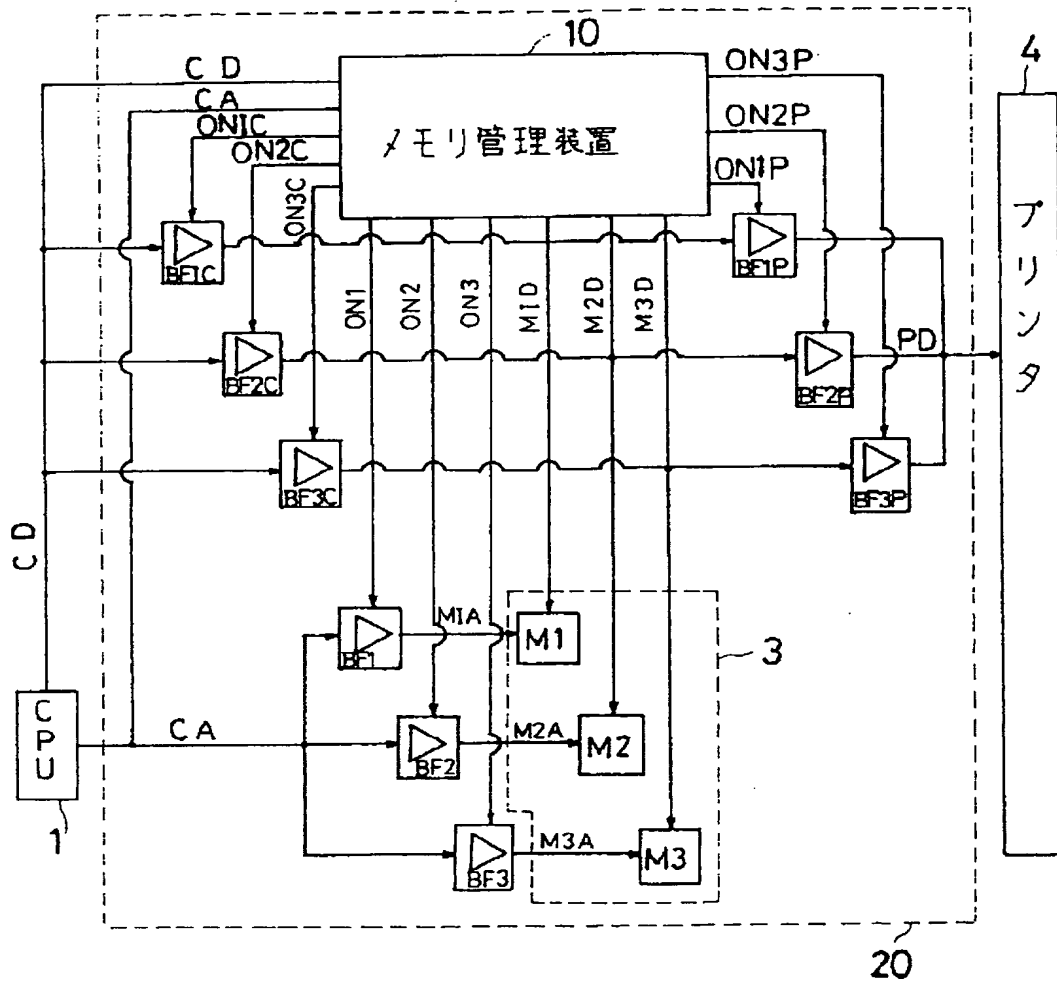
【図 9】

A	あ	1個の展開用メモリ で展開できる量
B	い う	
C	え	,
D	お か	,

【図 10】

アドレス	
4000H	M 3
3000H	M 2
2000H	M 1
1000H	

【図 3】



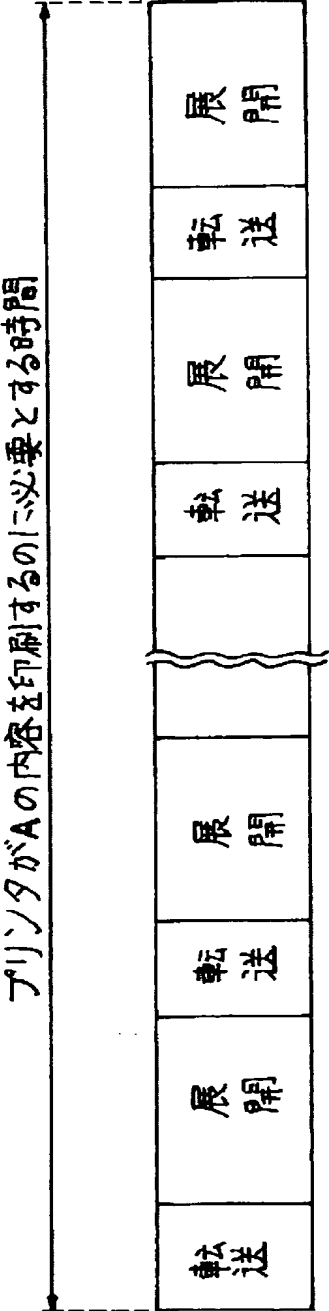
【図 14】

文字コード	描画位置		文字サイズ	
	x	y	縦	横
‘あ’	100	50	150	150
‘い’	300	200	150	150
‘う’	500	400	100	100
‘え’	100	700	150	150
‘お’	300	850	200	200
‘か’	600	1000	150	150

【図 6】

時刻	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆	t ₇	t ₈
メモリ状態		A	A	B	B	C	C	D	D
プリンタ出力状態			A	A	A B	A B	A B C	A B C	A B C D
CPUの動作状態	<div><div>Aの内容をメモリから プリンタへ メモリへ展開 転送</div><div>Bの内容をメモリから プリンタへ メモリへ展開 転送</div><div>Cの内容をメモリから プリンタへ メモリへ展開 転送</div><div>Dの内容をメモリから プリンタへ メモリへ展開 転送</div></div>								

【図 7】



【図8】

時刻	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅
メモリ状態		<div>A X</div>	<div>A B</div>	<div>C B</div>	<div>C D</div>	<div>X D</div>
プリンタ出力状態			<div>A</div>	<div>A B</div>	<div>A B C</div>	<div>A B C D</div>
CPUの動作状態	<div> <div>Aの内容をメモリへ展開</div> <div>Bの内容をメモリへ展開</div> <div>Cの内容をメモリへ展開</div> <div>Dの内容をメモリへ展開</div> </div> <div> <div>Aの内容をメモリからプリンタへ転送</div> <div>Bの内容をメモリからプリンタへ転送</div> <div>Cの内容をメモリからプリンタへ転送</div> <div>Dの内容をメモリからプリンタへ転送</div> </div>					

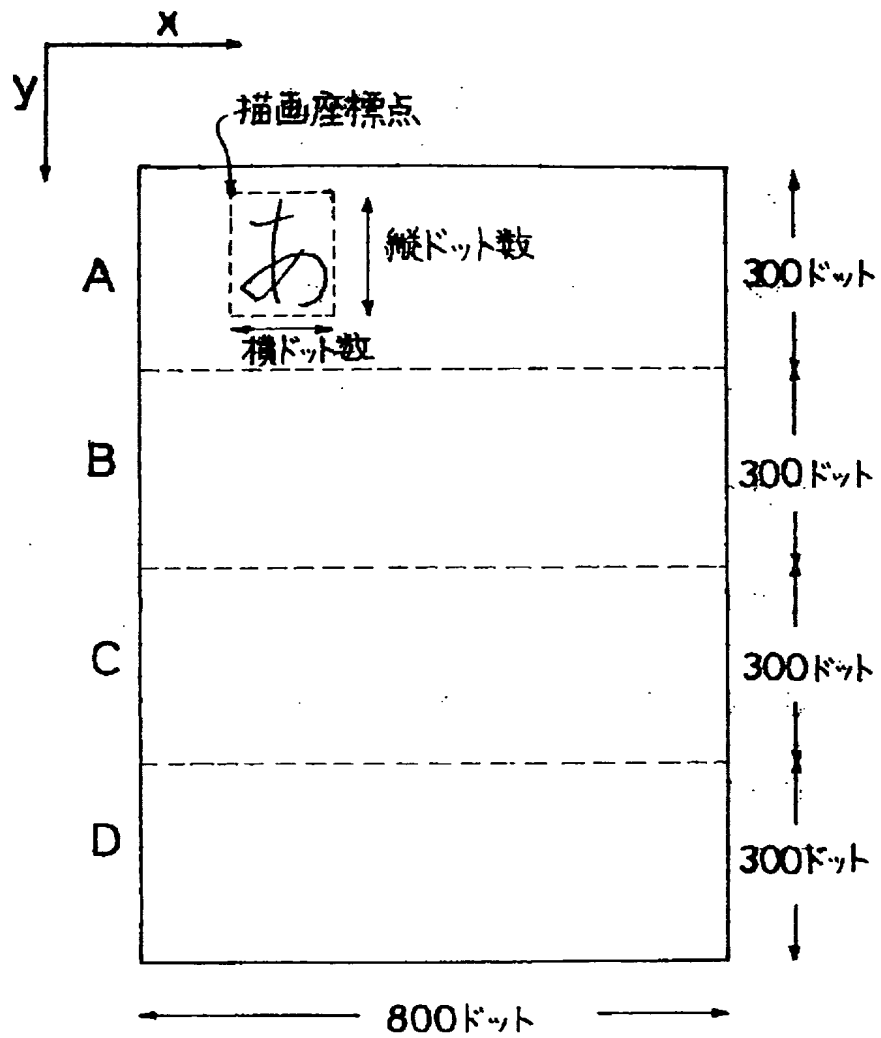
【図11】

		CPUからのアドレス		
		M1に対するアドレス	M2に対するアドレス	M3に対するアドレス
CPUの制御信号の状態	リード状態	ON1とON1Pのみをアクティブにする.	ON2とON2Pのみをアクティブにする	ON3とON3Pのみをアクティブにする.
	ライト状態	ON1とON1Cのみをアクティブにする.	ON2とON2Cのみをアクティブにする	ON3とON3Cのみをアクティブにする.

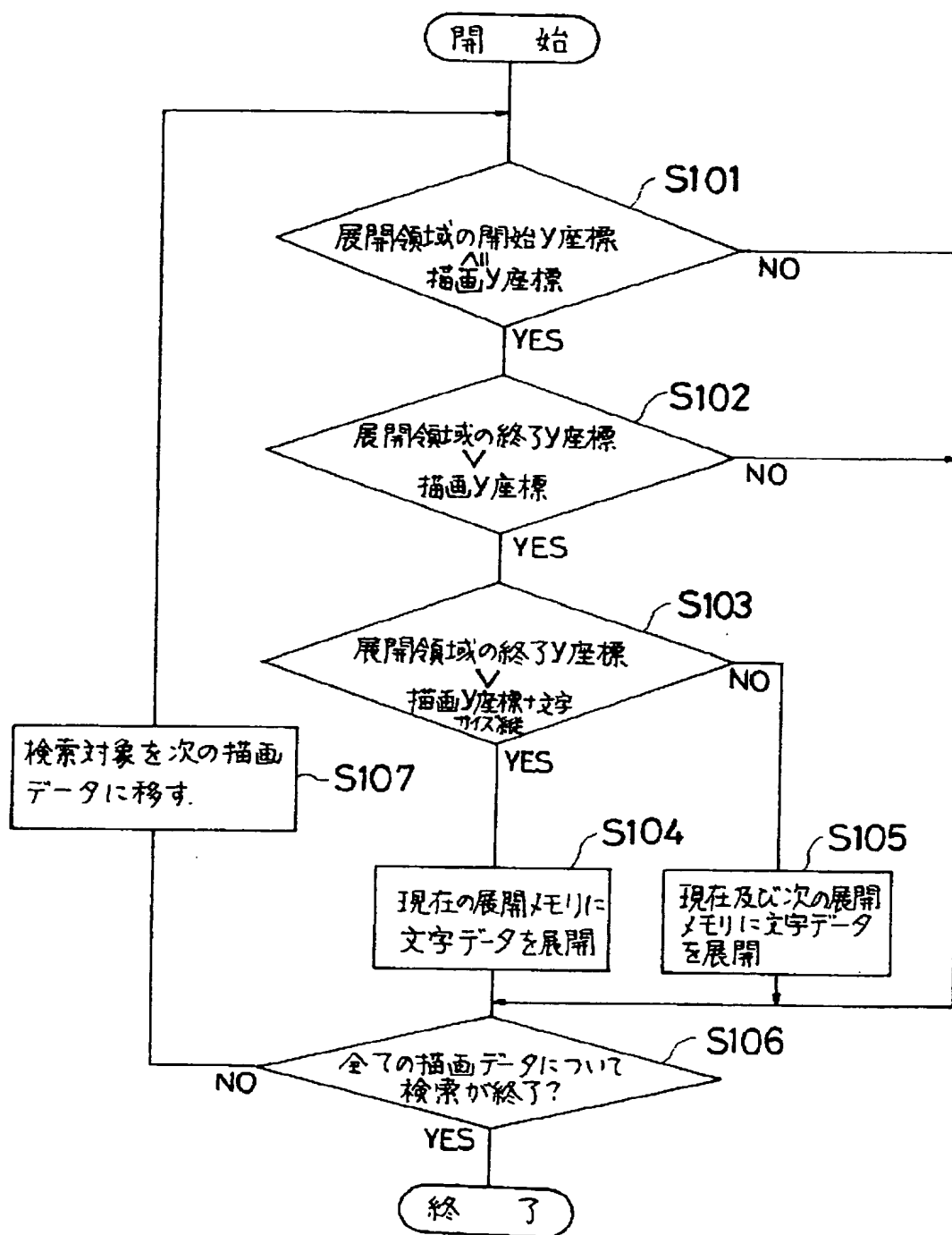
注：いずれでもない場合は何もアクティブにしない。

時刻	t ₀	t ₁	t ₂	t ₃	t ₄	t ₅	t ₆
メモリ状態	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M1</div> <div style="border: 1px solid black; padding: 2px;">A</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M2</div> <div style="border: 1px solid black; padding: 2px;">AB</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M3</div> <div style="border: 1px solid black; padding: 2px;"></div> </div>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M1</div> <div style="border: 1px solid black; padding: 2px;">A</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M2</div> <div style="border: 1px solid black; padding: 2px;">B</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M3</div> <div style="border: 1px solid black; padding: 2px;">BC</div> </div>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M1</div> <div style="border: 1px solid black; padding: 2px;">CD</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M2</div> <div style="border: 1px solid black; padding: 2px;">B</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M3</div> <div style="border: 1px solid black; padding: 2px;">C</div> </div>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M1</div> <div style="border: 1px solid black; padding: 2px;">D</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M2</div> <div style="border: 1px solid black; padding: 2px;">B</div> </div> <div style="display: flex; flex-direction: column; align-items: center;"> <div style="border: 1px solid black; padding: 2px;">M3</div> <div style="border: 1px solid black; padding: 2px;">C</div> </div>			
プリンタ出力状態		<div style="border: 1px solid black; padding: 2px; text-align: center;">A</div>	<div style="border: 1px solid black; padding: 2px; text-align: center;">A</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">B</div>	<div style="border: 1px solid black; padding: 2px; text-align: center;">A</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">B</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">C</div>	<div style="border: 1px solid black; padding: 2px; text-align: center;">A</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">B</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">C</div> <div style="border: 1px solid black; padding: 2px; text-align: center;">D</div>		
CPUの動作状態	A, ABの内容を (B-AB)BCの内容を (C-BC)CDの内容を (D-CD)の内容を M1, M2に展開 M2, M3に展開 M3, M1に展開 M1に展開						Aの内容を M1からプリンタへ転送 Bの内容を M2からプリンタへ転送 Cの内容を M3からプリンタへ転送 Dの内容を M4からプリンタへ転送

【図13】



【図15】



フロントページの続き

(51)Int.Cl.⁵
B 4 1 J 5/30識別記号 庁内整理番号
Z 8907-2C

F I

技術表示箇所

(72)発明者 沼尾 孝次
大阪府大阪市阿倍野区長池町22番22号 シ
ャープ株式会社内